2019 TÜRKİYESİNDE PERİYODİK CETVEL KONUSUNA DİJİTAL İÇERİK GELİŞTİRME YAKLAŞIMIYLA TEKNOLOJİK BİR BAKIŞ

1.GİRİŞ

Dimitri Mendeleyev periyodik tablonun modern halini insanlığa kazandırmış olup, tabloyu bulurken başından geçen olaylar mizansen olarak kurgulanmıştır. Öğrenciler açısından gelişen teknoloji ve bilgisayarların hayatımızda ki yeri düşünüldüğünde gerek akıllı telefonlarla gerekse internet aracılığıyla zamanımızın büyük bir kısmını ekran başında geçirmekteyiz.Periyodik tablo şüphesiz ilgili veya ilgisiz tüm öğrencilerin sadece işlerini görene kadar, genellikle ezbere dayalı olarak öğrendikleri ve işlerinin bittiğinde de zihinlerinde ket vurduğu bir konu olarak görülmekle beraber; bu projemizle hazırlamış ve kurgulamış olduğumuz heyecanlı ve bir o kadar da zevkli şekilde öğrencilerin eğlenerek, yaşayarak öğrenebilecekleri bir platformda öğrenmelerine katkıda bulunmak, öğrenci ile konu arasında lokomotif olarak, dersin daha zevkli ve anlaşılabilirliğini artırmak adına katkı sağlamak projenin çıkış noktasını oluşturmaktadır.

Anahtar Kelimeler: Periyodik Tablo, Dimitri Mendeleyev, Oyun, Yaşayarak Öğrenme, Görsel Sunu

2.ÖZET

Kahramanımız Dimitri evinde asılı olan periyodik tabloya incelerken tabloda mantık hatalarıyla karşılaşır.Bu hatalar zincirini çözmek için bili dünyasının kapısını aralar ve macera başlar...

18 elementten yola çıkarak oyunumuzda 6 ayrı bölüm ve her bölümde 3 ayrı elemente ulaşılacak heyecan dolu bir yolculuk başlar.Kahramanımıza bu yolda değişik ve birbirinden farklı ip uçları sunulacak.Tabii ki kahramanımız bu ip uçlarının farkına varırsa.....

Kahramanımıza sunulan ip uçları ve değişik tecrübelerle, oyun içerisinde gerek deney yapacak, gerekse bilgi notlarından kendisini elementlere götürecek yolu bulmaya çalışacaktır.18 elementi bulduğunda kapısı beklenmedik bir misafir tarafından çalınacaktır.Gizemli misafir Henry, kahramanımıza yardım edebileceğini söylemektedir.Asıl şimdi kahramanımız Henry' e güvenerek kapısını açmalımıdır, yoksa kendi yolundan devam mı etmelidir?

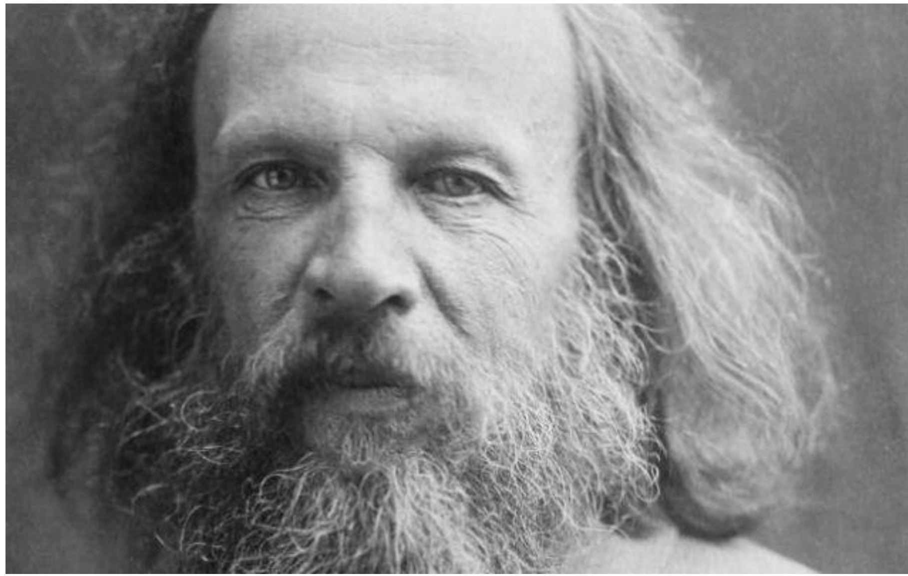
2.1- PROJENİN AMACI

Periyodik tablonun modern hale getirilmesinde 32 yaşında genç bir profesörün sanal alemde vücut bulmuş halini öğrenciye yaşatarak, kendisini dimitri mendeleyev olarak hissederek, 18 elementi bulma konusunda çözeceği formüller, kısa yollar,işaretler...vb gibi ip uçlarıyla tabloyu tamamlayarak, insanlığa hediye etmesini kurgulamak, Hali Hazırda ki sıradan eğitim ve öğretim materyallerinin aksine öğrencilerin yaşları dikkate alındığında ilgi ve istekleri doğrultusunda konuyu yaşayarak öğrenmelerini sağlamak.

2.2 - Dimitri Ivanovich Mendeleyev Kimdir?

St. Petersburg Üniversitesi'nde 1869 yılında, düzenlilikleri araştıran henüz daha 32 yaşında olan genç bir kimya profesörü, Dimitri Ivanovich Mendeleyev, mevcut olan elementleri belirli bir düzene göre sıraya koyan bir tablo yapılırsa birebir özellikleri gösteren elementlerin, alt alta geleceğini fark eder. Üstelik o dönemde elementlerin çoğunun keşfi gerçekleşmemiş, bazıları ise tam olarak bulunmamıştır.

Devrim niteliğinde ki bir öneme sahip olan buluş, başlangıçta pek çok buluşla aynı hüsrana uğramış, önemi yeterince kavranamamıştır. Fakat varlıkları ve özellikleri düşünülerek, tablodaki yerleri boş olarak bırakılmış olan elementlerin daha sonradan bulunması ve tabloya eklenmesi ile keşfin öneminin anlaşılmasına kaynaklık etmiştir.



Resim : 1 (Dimitri Mendeleyev)

Elementlerin Düzenlenmesi

Tek çeşit atomdan oluşan elementlerin tanımlanması, 1700'lü yılların ortalarına rastlar. Yüzyıl kadar sonra elementler, çeşitli semboller ve akronimlerle ifade edilmeye başlanmıştır. Ancak daha henüz ortak bir sözlük oluşmamıştır. 1869 yılında, Rus kimyager Dimitri Mendeleyev, bilinen elementlerle kendisi bir tablo tasarlar. Bütün elementleri içeren bu temel element tablosu, periyodik tablo olarak bilinmektedir. Periyodik tablonun esas ilginç kısmı, henüz keşfedilmemiş olan elementlerin hem varlıklarını hem atomik kütlelerini hem de kimyasal özelliklerini öngörmüş olmasıdır. Yeni elementlerin keşfi ile tablodaki boşluklar Mendeleyev'in öngördüğü şekilde tamamlanmıştır.

Mendeleyev'in Hayatı ve Periyodik Tablonun Keşfi

Mendeleyev'in dahi bir bilim adamı olma hikayesi, zorlu bir çocuklukla başlar. Mendeleyev, 1834 yılında Sibirya'nın batısında on (belki de daha fazla) çocuklu bir ailede dünyaya gelir. Ailenin birbiri ardına pek çok büyük sorun yaşadığı bilinmektedir. Öğretmen olan baba kör olmuş, annenin çalıştığı fabrika yanmıştır.

Büyük ekonomik sıkıntılar içinde olmalarına rağmen anne, Dimitri'nin çok parlak bir zekası olduğunu fark etmiş, iyi bir eğitim alması gerektiğine inanmıştır. Eğitim alabilmesi için oğlunu önce Moskova'ya, ardından St. Petersburg'a götürmüş ancak kısa bir süre sonra tüberkülozdan ölmüştür. Üstelik aynı hastalık bir süre sonra babasının ve kardeşlerinden en az birinin daha ölümüne yol açmıştır.

Mendeleyev'in Moskova'da başlayan eğitim hayatı, Avrupa'da yaptığı araştırmalarla devam etmiştir. 1861'de Rusya'ya geri dönmüş ve St. Petersburg Teknik Enstitüsü'nde ders vermeye başlamıştır. Popüler bir profesör olduğunda sadece 27 yaşındadır ve görünümüne de yansıyan çok ilginç bir kişiliği vardır.

Grup →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Periyot ↓	1A	2A	3B	4B	5B	6B	7B	8B	8B	8B	1B	2B	3A	4A	5A	6A	7A	8A
1	1 H																	2 He
2	3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
3	11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
4	19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
5	37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
6	55 Cs	56 Ba		72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
7	87 Fr	88 Ra		104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Uut	114 Fl	115 Uup	116 Lv	117 Uus	118 Uuo

Lantanidler	57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu
Aktinidler	89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr

Alkali Metaller | Toprak Alkali Metaller | Halojenler | Soygazlar

Metaller | Yarı metaller | Geçiş metalleri | Ametaller

Resim:2 (Periyodik Cetvel)

Organik Kimya alanında yazılmış bir kitap olmadığını fark edince; dönemin en önemli başvuru kaynağı olarak kabul edilen, canlılar ve karbon bileşikleri ile ilgili bir kitap yazmıştır (1861). Bir öğrencisinin gözlemine göre Mendeleyev, kitabının anlaşılma güçlüğü yarattığını fark etmiş ve bundan derin bir üzüntü duymuştur. Anlaşılma güçlüğünü, bilinen elementleri sınıflandıracak net bir sistemin bulunmamasına bağlamıştır. Ona göre sınıflandırma sisteminin olmaması, elementler arasındaki ilişkileri görecek bir çerçeveden yoksun olmak anlamına gelmektedir. Bunun üzerine inorganik kimya üzerinde çalışmaya başlar. İki ciltlik "Kimyanın İlkeleri" adlı eserini kaleme alır. Amacı ögeleri organize etmek ve açıklamaktır. İşe "tipik" elementler adını verdiği elementlerle başlar: Hidrojen, oksijen, azot, karbon vb. Bu elementlerin doğal bir düzen içinde olduğunu fark eder. Daha sonra atomik ağırlıkları düşük olan, diğer elementlerle kolayca reaksiyona girebilen ve doğada kolaylıkla bulunabilen elementleri listeler. Örgütleme ilkesi olarak, atom ağırlığını kullanmaya başlar. Bu çabalar tek başına net bir sistem oluşturmaya yetmez.

Her bir element için iskambil kağıdına benzer bir kart hazırlar. Kartları dizip bozarak sistemi keşfetmeye çalışır. Gece gündüz demeden çalışmasının sonunda, sistemin gizemini korumasını sağlayanın henüz keşfedilmemiş elementler olduğunu fark eder.

Hatta bir hikayeye göre, üç gün boyunca uyumadan çalışıp, element kartlarını yeniden ve yeniden dizdikten sonra uyuya kalmış ve rüyasında bir masanın üzerine yerleştirilmiş kartlarda periyodik cetveli görmüştür.

2.3 - Yazılım Nedir ?

Yazılım, bilgisayarın belirli bir işi yapabilmesi için tasarlanan komutların tümüdür. Kullanım amaçlarına göre çeşitli yazılım türleri mevcuttur. Yazılımlar genel olarak üç kısımda incelenebilir :

2.3.1-Sistem Yazılımları: Bilgisayar donanımlarını yöneten, uygulamaların çalışabilmesini sağlayan ve bu programlar için gerekli her türlü altyapıyı sağlayan yazılımlardır. İşletim sistemleri bu yazılım türüne örnek verilebilir.

2.3.2-Uygulama Geliştirme Yazılımları: Yazılım geliştiricilerin, son kullanıcıların ihtiyaçlarını karşılamak üzere, uygulamaları daha kusursuz geliştirmek, düzenlemek, hata payından kurtulmak, test etmek ve yayınlamak için kullandıkları yazılım türüdür. MS Visual Studio .NET bu türe örnek yazılımlardan bir kaçıdır.

2.3.3-Uygulamalar: Son kullanıcıların kendileri için belirledikleri ve belirli görevleri yerine getirmek amacı ile kullandıkları yazılımlardır. Uygulamalarda kendi içerisinde üç kısımda incelenebilir:

2.3.3.1-Windows Uygulamaları: Windows uygulamaları, işletim sistemi üzerine kurulan, arayüz ve içerik bakımından zengin uygulamalardır. Windows uygulamaları durum yönetimi, olaya dayalı programlama olayları olduğu için web uygulamalarına göre daha avantajlıdır. Windows uygulamalarına örnek olarak, MS Word, MS Live Messenger uygulamaları örnek verilebilir.

2.3.3.2-Web Uygulamaları: Bir sunucu tarafından yayınlanan bir web uygulamasını görüntülemek için istemci tarafından sadece bir uygulamayı çalıştırmak yeterlidir. Kurulum güçlükleri sunucu taraflı olduğu için yaşanmamaktadır. Windows uygulamalarına göre daha az avantajlıdırlar. Genellikle .NET uygulamaları ile yazılmaktadırlar.

2.3.3.3-Kurumsal Uygulamalar: Kurumsal uygulamalar, ölçeklenebilirlik, iş yükü ve güvenlik gibi ihtiyaçlara sahiptirler. Bu uygulamaların tek bir uygulama üzerinde çalıştırılması uygulamanın geliştirilme sürecini zorlaştırmaktadır ve performansı olumsuz yönde etkilemektedir. Bu sebeple farklı teknolojiler içeren kurumsal uygulamalar geliştirilmektedir. Bu uygulamalara örnek olarak MS Server 2008 R2 örnek gösterilebilir

3. PROJEDE KULLANILAN PROGRAMLAR VE MATERYALLER

3.1 - UNITY3D Programı

Unity3D Rusya da gelistirilen bir oyun motorudur. Aslinda cikis amaci oyunlari bilgisayarlara yuklemenize gerek kalmadan Unity Web Player ile internet uzerinden oynatabilmekti. Boylece hem siz yeni bir bilgisayar almaya ihtiyac duymayacaktiniz hem de korsan oyunlarin onune gecilmis olunacakti (Yapimci oyunu kendi sunucusundan oynattigi icin). Fakat Unity3D ekibi bununla yetinmedi ve her platforma destek verdi.

Unity3D nin 2012 yilinin aralik ayinda cikan son versiyonu 4.0 ile Unity artik Resmi olarak Desktop, Android, IOS, Flash Player, PS3, Web Player ve XBOX platformlari icin oyun yapmaniza imkan saglamaktadir.

Unity3D isminden de anlasilacagi gibi 3 boyutlu bir oyun motorudur. Unity ile 2 boyutlu oyun tasarlamaniz tavsiye edilmez

Unity3D oyun motoru Unity Engines tarafindan C/C++ ile gelistirilmistir. Unity, 4.0 surumu ile C#, JavaScript, Boo ve DirectX dilleri ile oyun gelistirmenize destek vermektedir.

Diğer oyun motorlarından farkı nedir?

*Unity3D' nin diğer oyun motorlarından farkı ilk olarak diğer oyun motorlarına göre çok daha esnek ve ucuz olmasıdır. Bu hem Unity3D' nin hızlı gelişmesine hem de oyun yapımcıları tarafından tercih edilmesini sağlamaktadır.

* Diğer bir farkı ise Unity3D tek tıklama ile PC için yaptığınız oyunu diğer platformlara çevirmenize olanak sağlar. Bu özelliği sayesinde Android icin yaptığınız bir oyunu kısa zaman diliminde IOS versiyonuna dönüştürebilirsiniz..

*Unity3D ' nin diğer farklı yönü Unreal gibi gelismis oyun motorlarının sunduğu fizik motoru, shader yazılımı, animation gibi farklı alt yapıları sizin kullanımınıza sunar.

Niçin Unity3D Kullandık?

- Diğer oyun motorlarının aksine bilgisayar ortamından 8 farklı PS3, iOS, Wii gibi platformlarda aynı anda ileri sürülme imkanı sunar.
- Kullanımı her yaştan kullanıcının algılayacağı kadar basitleştirilmeye çalışmıştır.
- Yaptığınız oyun veya üründen 100.000'den üzeri bir gelir elde etmezseniz sizin paranıza hiçbir şekilde ortak olmuyor.Telif istemiyor.
- Girişimci oyun geliştiricileri ve acemiler için ideal bir program

- CryEngine ve ya Unreal Engine tarzı gelişmiş ve profesyonel oyun motorlarına geçmeden önce kullanılırsa ciddi bir şekilde zaman ve ekonomik avantaj sağlar.
- Programlama dilleri olan C# ve JavaScript çoğu programcı tarafından yaygın olarak kullanıldığı için yeteri kadar bilgi ve videoya ulaşabilirsiniz.
- Web Player ile yükleme gereksinimini minimum seviyeye indirger.

3.2 - Visual Studio Programı

Visual Studio, Microsoft tarafından üretilen ve konsollar, grafik kullanıcı ara yüzleri, Windows formları, Web servisleri ya da Web uygulamaları oluşturmak için kullanılan bir IDE'dir. Visual Studio programı içerisinde yalnızca Microsoft Windows tarafından desteklenen yerel kodlar kullanılmaktadır.

Visual Studio'nun bir nevi yazılım üretmek için yazılım olduğunu söyleyebiliriz. Visual Studio yazılımları elbette kendisi kodlamıyor ancak geliştiricilere ileri seviye özellikleri sayesinde yardımcı olarak daha kısa süre içerisinde programlarını hazırlama imkanı sunuyor. Visual Studio programını kullanarak bilgisayar yazılımları, web uygulamaları ve web servislerini çok daha hızlı ve kolay bir şekilde hazırlayabilirsiniz.

Visual Studio son derece fonksiyonel ve şık ara yüz tasarımına sahiptir. Bu fonksiyonel arayüzle birlikte Visual Studio yazılımcıların yükünü hafifletecek bir kod editörü, debugger, GUI tasarlama aracı, veri tabanı şema tasarım aracı ve öncül revizyon kontrol sistemlerini sunuyor. Visual Studio yazılımın ticari amaçla kullanılan ticari program şeklinin yanı sıra topluluk bazlı olarak geliştirilen ücretsiz sürümü de bulunmaktadır.

Visual Studio her ne kadar bir geliştirme ortamı olsa dahi Eclipse gibi bu ortam içerisinde birçok programlama dilini kullanamıyor, sadece Visual Studio tarafından desteklenen diller üzerinden çalışabiliyorsunuz. Visual Studio yazılımının desteklediği programlama dilleriyse şunlardır;

- C
- C#
- F#
- Python
- HTML/XHTML/CSS
- JavaScript
- Visual Basic .NET
- C++
- Fossil
- M

3.3 - Wondershare Filmora Editor Programı

Wondershare Filmora Video Editor, video düzenleme sürecini olabildiğince basit hale getirmek için tasarlanmış, güçlü ve çok yönlü bir yazılım parçası olarak tasarlanmış bir programdır.

4. YÖNTEM

4.1- Projenin Tespiti

*Konunun kurgulanması, senaryo hazırlanması

*Sorunun çözümlenmesine dair hangi programların kullanılabileceğinin araştırlması.

*Görüntü, Ses, Sahne tasarımlarının gerçekçi ve oyunun sahnelerindeki akışın bir birleriyle olan ilişkilerini max. uyumlulukla çalıştırma hedefi.

4.2 - Proje yapım aşamaları

Oyun hakkında fikir ararken, insanların fen konusunda en çok takıldığı noktalardan birinin periyodik tablo olduğunun fark edilmesiyle beraber bu periyodik tabloyu daha zevkli ve anlaşılır şekilde öğretebilecek bir oyun yapma fikri belirdi,

İlk olarak periyodik tablonun tüm geçmişi araştırıldı ve periyodik tabloda emeği geçen tüm bilim adamlarından en çok geliştiren bilim adamının oyunun ana karakteri (Dimitri Mendeleyev) yapılmasına karar verildi. Oyun ile alakalı tüm senaryo ve mizansen kurgulandıktan sonra oyunun başından sonuna sahnelere karar verdikten sonra oyun geliştirmeye başlandı.

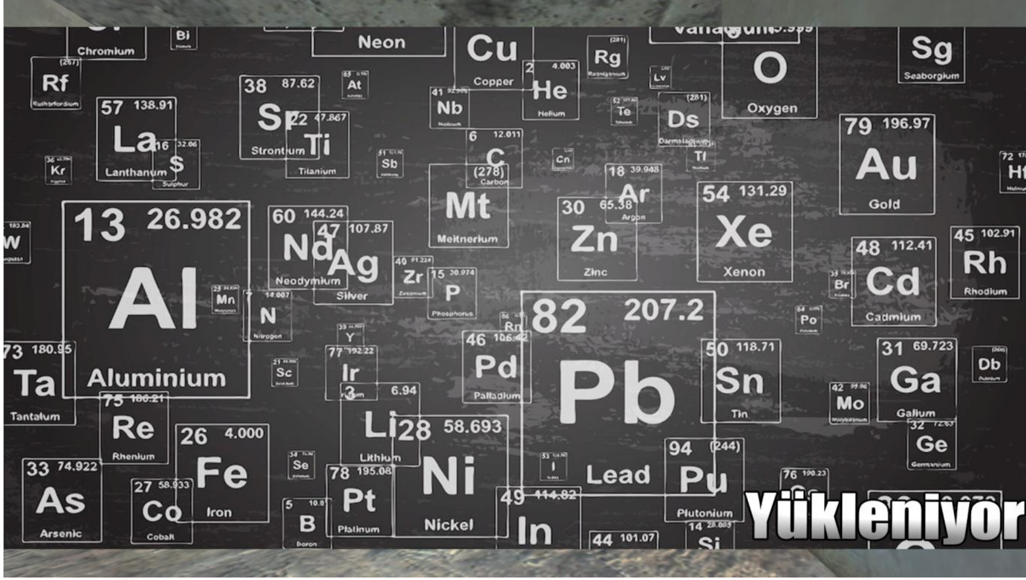
Daha sonra öğrencinin önceden C# kodlama dilini kullanarak yazılım geliştirdiği için bu kodlama dilini kullanan bir oyun motoru bulmaya karar verildi ve bu C# kodlama dili ile oyunu geliştirebileceği "Unity3D" isimli oyun motorunu kullanmasının daha sağlıklı olduğu tespit edildi.

Akabinde C# kodlama dilinde birbirinden farklı işlere yarayan kodları; ‘’katagorilere’’ ayırarak farklı renklere ayırması gibi özelliklerden uzun satırlı kodlarda; kodları daha kolay ayırt edebilmesi sayesinde Visual Studio 2017 yazılımını kodların yazıldığı program olarak kullanmaya karar verildi.

Örnek: *Public Bool kutu; kodunda Public ve Bool koyu mavi renkte yazılmışken kutu benim oluşturduğum ve kütüphanede bulunmayan bir obje olduğu için beyaz renk aldı.*

Visual Studio: `public bool kutu;`

Diğer kod yazılabilen yazılımlar: `Public Bool kutu;` , `Public bool kutu;`



Resim: 3 (oyun açılış görüntüsü)

Devamında son olarak oyun içindeki ses dosyalarını düzenleyebilmek için kullanımı basit olması nedeni ile Wondershare Filmora Editör yazılımından faydalanıldı.

5. BULGULAR VE GERÇEKLEŞME

5.1 – Kodlamaya Örnek(*)

using UnityEngine.UI;//UnityEngine isimli kod kütüphanesini oyuna ekle(Ekranda çıkacak objeleri düzenlemeye yarar).

using UnityEngine.SceneManagement;//Sonraki sahneye geçiş gibi kodları barındıran kütüphaneyi oyuna ekle.

using UnityEngine;// Unity nin ana kütüphanesini oyuna ekle.

public class AnaDosyaS3 : MonoBehaviour

{

private float mesafe = AnaDosya.mesafe;// Mesafe değişkenini AnaDosya isimli dosyadaki mesafe değişkenine eşitle.

public Image crosshair;// Ekranın Ortasındaki Nokta.

public AudioSource seskaynagi;// sesin oynatılacağı nesne yani ses kaynağı.

public AudioSource seskaynagi1;// sesin oynatılacağı nesne yani ses kaynağı.

public AudioClip gerisayim;// Geri Sayım Sesi.

public AudioClip patlamases;// Hidrojen Tepkimesindeki Patlama Sesi.

public AudioClip diyalog1;// Diyalog ses dosyası 1.

public AudioClip diyalog2;// Diyalog ses dosyası 2.

public AudioClip diyalog3;// Diyalog ses dosyası 3.

public AudioClip hedef;// Oyuncu 1. Hedefi Eline Aldığında Oynatılacak Ses Dosyası.

public AudioClip hedef1;// Oyuncu 2. Hedefi Eline Aldığında Oynatılacak Ses Dosyası.

public AudioClip iseyaramaz;// Oyuncu işine yaramayacak bir şey aldığında oynatılacak ses dosyası.

public AudioClip diyalog4;// Diyalog ses dosyası 4.

public AudioClip diyalog5;// Diyalog ses dosyası 5.

public PlayerScript oyuncu;// PlayerScript olan bir değişkeni oyuncu olarak belirler.

public MouseAyari oyuncu1;// Mouse'yi oynatmaya yarayan eklenti olan nesneyi belirler.

public bool bombaizin = false;// hidrojen tepkimesine izin verir veya yasaklar.

public GameObject kaya1;// Kaya nesnesini belirler.

private bool keycodekapaliyetki = false;// Kapı Kilidini Açma Yetkisi Verir.

public GameObject patlamaSFX;//Hidrojen ile alev tepkimesinde oluşacak patlama efekti.

public GameObject keypad;// Kapı şifresini girmeye yarayan nesne.

[SerializeField] Text gerisayimtext;// Geri sayım başladığında çıkan yazı "Tepkimeye kalan süre".

[SerializeField] Text gerisayimtext1;// Geri sayım başladığında çıkan yazı "5, 4, 3, 2, 1".

private float gerisayim1;// geri sayımın sayı değişkeni.

public GameObject NotDosya;// Not Objesi.

public Image NotEkrani;// Not ele alındığında çıkan not ekranı.

public Sprite afis1;// Not ekranında çıkacak notun resim dosyası.

public Rigidbody kaya1_1;// Yerçekiminden etkilenen kaya objesi 1.

public Rigidbody kaya1_2;// Yerçekiminden etkilenen kaya objesi 2.

public Rigidbody kaya1_3;// Yerçekiminden etkilenen kaya objesi 3.

public Rigidbody kaya1_4;// Yerçekiminden etkilenen kaya objesi 4.

public Rigidbody kaya1_5;// Yerçekiminden etkilenen kaya objesi 5.

public Rigidbody kaya1_6;// Yerçekiminden etkilenen kaya objesi 6.

public Rigidbody kaya1_7;// Yerçekiminden etkilenen kaya objesi 7.

public GameObject hidrogen;// hidrojen gazı

```
public bool keycodeacik;
public GameObject gerisayimekran;
public GameObject keycode;
private bool notacikkapali;
private bool notacikkapaliyetki;
private bool kaya;
private bool patladi;
private bool elektrik;
// Update is called once per frame
void Update()
{
    Vector3 ileri = transform.TransformDirection(Vector3.forward);// ileri değişkeninin değerini belirler.
    RaycastHit hit;// oyuncunun baktığı yönü kaydeder

    if (gerisayim1 <= 4)// Eğer geri sayım 4 den düşük ise ekranın ortasındaki noktayı beyaz renk yap.
    {
        gerisayimtext.color = Color.white;
        gerisayimtext1.color = Color.white;
    }
    if (gerisayim1 <= 3)// Eğer geri sayım 3 den düşük ise ekranın ortasındaki noktayı kırmızı renk yap.
    {
        gerisayimtext.color = Color.red;
        gerisayimtext1.color = Color.red;
    }
    if (gerisayim1 <= 2)// Eğer geri sayım 4 den düşük ise ekranın ortasındaki noktayı beyaz renk yap.
    {
        gerisayimtext.color = Color.white;
        gerisayimtext1.color = Color.white;
    }
```

```
if (gerisayim1 <= 1)// Eğer geri sayım 1 den düşük ise ekranın ortasındaki noktayı kırmızı renk yap.
{
    gerisayimtext.color = Color.red;
    gerisayimtext1.color = Color.red;
}
if (gerisayim1 <= 0)// Eğer geri sayım 0 den düşük ise veya 0 ise geri sayımı devre dışı bırak.
{
    gerisayimekran.SetActive(false);
}
if (Physics.Raycast(transform.position, ileri, out hit))
{
    if (hit.collider.gameObject.tag == "Untagged")// Eğer objenin etiketi yoksa ekranın ortasındaki nokta beyaz renk olsun.
    {
        crosshair.color = Color.white;
    }
}
if (Physics.Raycast(transform.position, ileri, out hit, mesafe))
{
    if (hit.distance <= mesafe && hit.collider.gameObject.tag == "keycode" & Playsound.sifregirildi == false)// eğer oyuncu objeye mesafe değerinden daha yakın ise, objenin etiketi keycode ise ve PlaySound dosyasındaki sifregirildi kodu devre dışı ise bu kodu oynat.
    {
        crosshair.color = Color.green;//Ekranın ortasındaki noktayı yeşil renk yap
        if (Input.GetKeyDown(AnaDosya.EsyaAlma))// Eğer oyuncu anadosya dosyasındaki EsyaAlma isimli tuşa(E) basarsa bu kodları oynat.
        {
            crosshair.color = Color.white;//Ekranın ortasındaki noktayı Beyaz renk yap.
            oyuncu.enabled = false;// oyuncu dosyasını devredışı bırak.
            oyuncu1.enabled = false;// oyuncu1 dosyasını devredışı bırak.
            keycodeacik = true;// keycodeacik isimli değişkeni "doğru" yap.
            keycode.SetActive(true);// keycode isimli değişkeni görünür hale getir.
```

Cursor.lockState = CursorLockMode.None;// Mouse imlecinin hareketine izin ver.

Cursor.visible = true;// Mouse imlecini görünür hale getir.

}

}

if (hit.distance <= mesafe && hit.collider.gameObject.tag == "iseyaramaz")// eğer oyuncu nesneye mesafe değerinden daha yakınsa ve nesnenin etiketi "iseyaramaz" ise bu komutu oynat.

{

crosshair.color = Color.green;// Ekranın ortasındaki noktayı yeşil renk yap.

if (Input.GetKeyDown(AnaDosya.EsyaAlma))// Eğer oyuncu AnaDosya isimli dosyanın içindeki EsyaAlma isimli tuşa "E" basarsa bu komutu oynat.

{

crosshair.color = Color.white;//Ekranın ortasındaki noktayı beyaz renk yap.

seskaynagi.clip = iseyaramaz;// Seskaynağının varsayılan ses dosyasını iseyaramaz ses dosyası olarak değiştir.

seskaynagi.Play();// seskaynağındaki varsayılan ses dosyasını oynat.

}

}

if (hit.distance <= mesafe && hit.collider.gameObject.tag == "not1")// Eğer oyuncu objeye mesafe değerinden daha yakınsa ve not1 isimli dosyayı açarsa bu komutu oynat.

{

crosshair.color = Color.green;//Ekranın ortasındaki noktayı yeşil renk yap.

if (Input.GetKeyDown(AnaDosya.EsyaAlma))// Eğer oyuncu AnaDosya isimli dosyanın içindeki EsyaAlma isimli tuşa "E" basarsa bu komutu oynat.

{

crosshair.color = Color.white;//Ekranın ortasındaki noktayı beyaz renk yap.

hit.collider.gameObject.SetActive(false);// Oyuncunun EsyaAlma(E) tuşuna bastığı objeyi devre dışı bırak.

notacikkapali = !notacikkapali;// notacikkapali değerini tam tersi yap.(açık ise kapalı, kapalı ise açık).

notacikkapaliyetki = true;//notacikkapaliyetki yi doğru yap yani notu açma yetkisi ver.

NotEkrani.sprite = afis1;//Not Ekranının Resim Dosyasını afis1 Olarak Belirle.

```
}
}
if (kaya == true & elektrik == true & patladi == false)//Eğer kaya değeri açık, Elektrik değeri açık ve patladi değeri kapalı ise bu komutu oynat.
{
    crosshair.color = Color.white;//Ekranın ortasındaki noktayı beyaz renk yap.
    Invoke("tuzakpatlama", 5f);//Kayapatlama ismindeki kod dizisini(void) oynat.
    seskaynagi1.clip = diyalog4;//Ses kaynağı 1 in varsayılan ses dosyasını diyalog4 yap.
    seskaynagi1.Play();//Ses kaynağı 1 deki varsayılan ses dosyasını oynat.
    patladi = true;// patladi değişkenini doğru yap.
}
// NOTLAR
if (Input.GetKeyDown(AnaDosya.NotGosterme))//Eğer oyuncu AnaDosya dosyası içindeki NotGosterme(TAB) tuşuna basarsa bu komutu oynat.
{
    keycodekapaliyetki = true;// keycodekapaliyetki değişkenini 1 yap.
}
if (Input.GetKeyDown(AnaDosya.NotGosterme))// Eğer oyuncu AnaDosya dosyası içindeki NotGosterme(TAB) tuşuna basarsa bu komutu oynat.
{
    Cursor.lockState = CursorLockMode.Locked;//Fare imlecini ekranın ortasına kilitle.
    Cursor.visible = false;//Fare imlecinin görünürlüğünü kapat(görünmez yap).
    if (keycodeacik == false)//eğer keycodeacik değişkeni yanlış ise.
    {
        notacikkapali = !notacikkapali;// notacikkapali değişkeni tam tersi duruma gelsin(açık ise kapalıi kapalı ise açık).
        notacikkapaliyetki = true;//not açıp kapatma yetkisi verilsin.
    }
    if (keycodekapaliyetki == true)// eğer keycodekapali yetkisi var ise.
    {
        keycodeacik = false;//Keycode Açık değişkeni yanlış olsun.
        keycode.SetActive(false);//Ekranda duran keycode ekranı devredışı olsun.
```

```
                oyuncu.enabled = true;// oyuncu dosyasını aktif etsin.
                oyuncu1.enabled = true;// oyuncu1 dosyasını aktif etsin.
            }
        }
    }
}
public void patlamakapi()//patlamaKapi isimli kod yığını(voidi).
{
    seskaynagi.clip = gerisayim;//Ses kaynağının varsayılan ses dosyası gerisayim olsun.
    seskaynagi.Play();//Ses kaynağının varsayılan ses dosyası oynatılsın.
    seskaynagi1.clip = diyalog3;//Ses kaynağı 1'in varsayılan ses dosyası diyalog3 olsun.
    seskaynagi1.Play();//Ses kaynağı 1 in varsayılan ses dosyası oynatılsın
    Invoke("patlamakapi1", 10f);//10 saniye sonra patlamakapi1 kod yığınını(voidini) oynat.
}
public void patlamakapi1()
{
    seskaynagi.clip = patlamases;//Ses kaynağının varsayılan ses dosyası patlamases olsun.
    seskaynagi.Play();//Ses kaynağının varsayılan ses dosyası oynatılsın.
    keypad.SetActive(false);//Keypad objesi devre dışı bırakılsın.
    patlamaSFX.SetActive(true);//Patlama efekti aktif olsun.
}
}
```



Resim: 4 (oyundan bir sahne)



Resim: 5 (oyundan bir sahne)

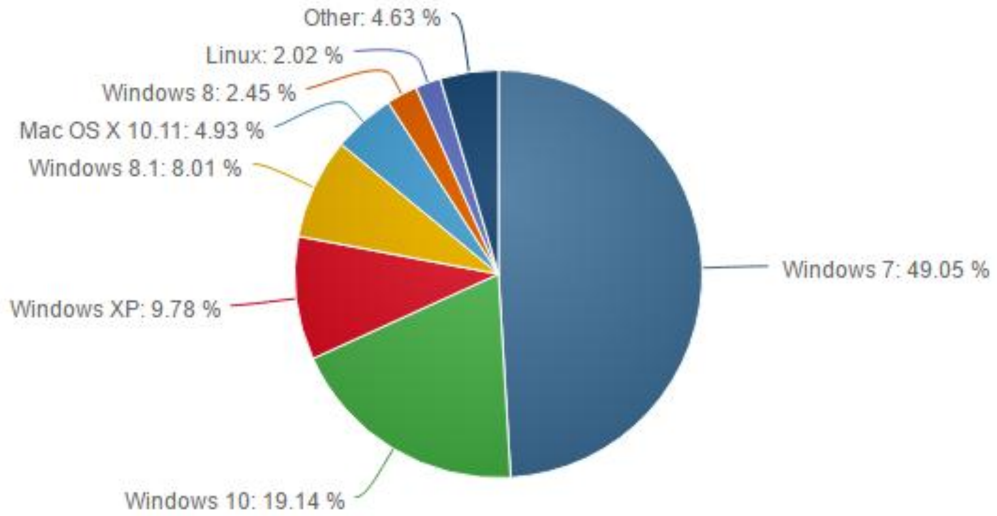
(*) Proje raporlama max.20 sayfadan oluşacağı için belirtilen kodlama oyunumuzun sadece 3. Bölümüne ait bir kısım.

5.2. - Yöntem

Tasarlanan oyun 3 temel program kullanılarak inşa edildi.Bunlar; Visual Studio, Unity3D ve sesler için Wondershare Filmora Video Editor programları.Visual Studio programında görüntüler ve sahneler hazırlandı, Unity3D programıyla oyunun kodlaması yapıldı,son olarak Wondershare Filmora Video Editor programıyla da ses ve görüntüler düzenlendi.

5.3 - Oyunumuzun Artıları

Hazırlamış olduğumuz oyunumuz Windows tabanlı işletim sistemlerinde (Vista, Windows7,8 ve üzeri) rahatça kullanılabilir.Dünyanın yaklaşık %93 ' ünün windows tabanlı işletim sistemi kullandığı düşünülürse ulaşılabilirliği açısından gayet tatminkar bir oran olduğu bellidir.



Resim: 6 (Dünyada kullanılan işletim sistemleri oran tablosu)

5.4 - Oyunumuzun Eksileri :

Oyunumuzun değişik platformlarda kullanılması proje aşamasında olduğundan şu an için sadece Windows tabanlı işletim sistemleriyle sınırlı kalmaktadır. İlerleyen dönemlerde; Android, Linux, İoS ve mobil cihazlar için sürümleri güncellenerek öğrencilerin kullanımına sunulabilir.



Resim: 7 (oyundan bir sahne)

6. SONUÇLAR VE TARTIŞMA

Yukarıda belirtildiği şekliyle yapılan oyun Orta okul 8. sınıf konusu olan periyodik tablo' nun öğrencilerin daha anlaşılabilir şekliyle günümüz teknolojisini de içine alarak öğrenme modellerine yeni ve farklı bir bakış açısı getirmek adına hazırlanmıştır. Görsellerden de anlaşılacağı üzere oyunumuzu öğrenciler oynarken deney yapacak, kendilerine verilen ip uçlarını çözecek, elementlerin özelliklerini öğrenecek, tepkimelerini ve bu tepkimeler sonucunda neler olacağını görecek.

Oyunumuz 6 bölümden oluşmakta ve her bölümde 3 farklı element, oyuncu (dimitri) tarafından çeşitli yollarla bulunarak 18 elemente tamamlanacak.Oyunumuzda 14 sahnemiz bulunmakta.

Oyun konsol tuşları istendiği takdirde ayarlar menüsünden, farklı tuşlar atanabilir.Tamamen oyuncunun kendi isteğine bırakılmıştır.



Resim: 8 (klavye tuşları)

7. ÖNERİLER

Oyunlar hiç şüphesiz günümüzde her hangi bir yaş kısıtlaması olmadan her yaştan bireylerin ilgi duyduğu, hatta uluslararası kategorilerde yarışmaların düzenlendiği ve hatırı sayılır izleyici kitlesinin olduğu inkar edilemez bir gerçektir. Diğer bir gerçek ise Milli Eğitim Bakanlığının 2023 Eğitim vizyonunda da belirtildiği üzere gelecek: öğrenme süreçlerinde dijital içerik ve beceri destekli dönüşüm yoluyla gerçekleşecek.Ezberden uzak; yaşayarak, deneyerek, araştırarak, görerek öğrenmenin temellerinin atıldığı 21. yüzyıl Türkiyesinde derslerde geçen konular animasyon, robotik, yazılım, tasarım ve bilişim gibi temel ve yardımcı yollarla anlatılacak.Sadece Bilgisayar kullanmanın günümüz şartlarında artık, çok büyük bir önemi kalmadığı, Eğitimde yeni yaklaşımlar ve STEM tarzı eğitim-öğretim şekli yerini almaya başlamıştır.

Bizde yapmış olduğumuz bu projeyle yeni eğitim öğretim düzeninde yer bulabilmek adına farkındalık oluşturmak istedik.Oyunumuz istendiği takdirde daha da geliştirilebir. Farklı konular eklenerek daha genel bir düzeye getirilebilir.



Resim: 9 (oyundan bir sahne)

8. KAYNAKÇA

https://medium.com/android-türkiye/unity-ile-mobil-oyun-hazırlama-1-8b91241673bd

http://www.mshowto.org/microsoft-visual-studio-nedir-ne-amacla-kullanilir.html

http://www.olaganustukanitlar.com/dimitri-mendeleyev-ve-periyodik-tablo/

http://www.temizkod.com/unity3d/

https://www.webtekno.com/yazilim/masaustu-isletim-sistemlerinin-kullanim-oranlari-aciklandi-h18404.html

https://wmaraci.com/nedir/visual-studio

https://wondershare-filmora-video-editor.tr.softonic.com/

Khan Academy– Dmitri Mendeleev-makale